Science of Computer Programming 16 (1991) 89-99
Elsevier

Book Reviews*

Jenny Preece and Laurie Keller, eds., *Human Computer Interaction: Selected Readings* (Prentice-Hall/Open University, Hemel Hempstead, United Kingdom, 1990), Price £28.50 (paperback), ISBN 0-13-444910-X.

The science of computer programming is a precise and increasingly well-understood one insofar as it deals with computers; but no computer is an island, and most computer systems include people. But whilst much rigour and effort may be applied to maximizing the efficiency of the mechanical parts of a system, the human parts are all too often left to adapt themselves to fit with the rest. But some tools are considerably easier for users to learn and use than others, and the human element may turn out to be the most critical one, affecting costs and ultimately the success of the system as a whole. The message of *Human Computer Interaction* is that we can measure and model human capabilities at the interface with machines, and apply that knowledge to system design: pleasing managers by achieving greater productivity, and pleasing users by providing a system which is more satisfying to work with.

This book is a collection of 21 papers by well-respected researchers, mostly published elsewhere throughout the eighties—the decade of user-friendliness. Compiled in conjunction with other course material by the Open University in the UK and the Netherlands, it is a stand-alone volume directed at undergraduates and working system designers, and serves as a good introduction to the field. Partitioned roughly by subject matter with a page of introduction to each section, the papers necessarily reflect different viewpoints and speak each in its own slightly different language; but the overall thrust is consistent, and every paper is readable and interesting. Most of the papers have good bibliographies.

The early chapters establish the history and scope of the topic, and then the standard of rigour is set by an analysis of graphics interaction techniques (pop-up versus pull-down, for example), followed by a plea for more creative use of suitable I/O hardware. However, throughout the book it is made clear that whilst the surface presentation layer is important, the study of *HCI* is by no means confined to that layer. To be usable, a machine must exhibit behaviour which humans can learn to understand; and of course its behaviour depends on every part of its architecture.

* Review copies of books which might be of interest to the readers of *Science of Computer Programming* should be sent to Prof. D. Bjoerner (address: see inside front cover). Proceedings of conferences will not normally be reviewed.

Therefore consideration of human factors must influence every part of a development, and not just be applied atop the design "like peanut butter" at the end.

The bulk of the content is taken up with sections on the application of rigorous techniques and theory adopted from the mainstream of cognitive psychology, and on their implications for design methodology. Software engineers are not noted for their qualifications in psychological experimental technique, and so here we are shown how. We see how useful models of human comprehension and capabilities can be built and used to guide the design of new systems, so as to predict and maximize their usability and learnability. We are shown how to measure these two quantities, and programmers' clients are encouraged to include them as evaluation criteria in requirements specifications.

Sometimes we only need rules of thumb ("recognition is better than recall" and the like); sometimes deeper analysis is appropriate. But psychology is not a precise or completely understood science, and there can be no guarantee that the most usable system will be achieved first time. (Contrast this with the functionality of the system, which we can—in theory—guarantee correct first time.) It must therefore be accepted as part of the development method for highly interactive systems, that every development path will include such stages as investigations of interactive styles, evaluations of performance, and subsequent updates in the light of those results.

The collection concludes with some specific methods and mechanisms: the User Software Engineering design methodology, and two papers on the use and design of User Interface Management subsystems.

I believe there must be few designers who could not benefit from reading this book. Very many of the systems with which we have to interact encourage errors, are clumsy to use, or are difficult to learn. User friendliness doesn't just mean using a mouse! In the days of batch processing, a compiler-writer only needed to be a specialist in compiling, and a theorem-prover-writer in theorem-proving. Now that these specialists are building interactive versions of their wares, they must recognize that there is a relevant science to be applied.

Alan Wills
Department of Computer Science
University of Manchester
Manchester, United Kingdom

Open Software Foundation, *OSF/Motif User's Guide* (Prentice-Hall, Hemel Hempstead, United Kingdom, 1990), Price £11.95 (paperback), ISBN 0-13-640509-6.

Open Software Foundation, *OSF/Motif Style Guide* (Prentice-Hall, Hemel Hempstead, United Kingdom, 1990), Price £12.95 (paperback), ISBN 0-13-640491-X.